ORC 65-33
DECEMBER 1965

AN EXTENSION OF GENERALIZED UPPER BOUNDED TECHNIQUES I FOR STRUCTURED LINEAR PROGRAMS

by

M. Sakarovitch
and
R. Saigal

OPERATIONS RESEARCH CENTER

COLLEGE OF ENGINEERING

UNIVERSITY OF CALIFORNIA - BERKELEY

AN EXTENSION OF GENERALIZED UPPER BOUNDED TECHNIQUES I FOR STRUCTURED LINEAR PROGRAMS

by

M. Sakarovitch and R. Saigal
Operations Research Center
University of California, Berkeley

December 1965

ORC 65-33

This research has been supported by the Army Research Office under Contract DA-31-124-ARO-D-331, and the National Institutes of Health under Grant GM-9606 and the National Science Foundation under Grant GP-4593 with the University of California. Reproduction in whole or in part is permitted for any purpose of the United States Government.

ABSTRACT

An algorithm is developed for solving a special structured linear program. The particular structure studied has a large number of blocks coupled together by a relatively few connecting equations. The method proposed is an extension of [2] and, from the basis, defines a working basis which is much smaller in size than the original. Two methods of updating the working basis are proposed.

INTRODUCTION

In this paper we propose a device for solving a large scale linear programming problem of a special configuration. The method proposed is an extension of "Generalized Upper Bounded Techniques - I" [2] to a larger class of problems.

The special configuration is the following:

$$\textcircled{1} \quad \begin{cases} \boxed{A'_o x_o + A'_1 x_1 + \qquad\qquad A'_L x_L} = b_o \\ \qquad \boxed{A_1 x_1} \qquad\qquad\qquad\quad = b_1 \\ \qquad\qquad\qquad\qquad \boxed{A_L x_L} = b_L \\ \gamma_o x_o + \gamma_1 x_1 + \qquad\qquad \gamma_L x_L = z(\min) \end{cases} .$$

x is a vector of N components which can be written $x = (x_o, x_1, \ldots, x_L)$ where x_i has n_i components and $N = n_o + n_1 + \ldots + n_L$.

A'_i is an $m_o \times n_i$ matrix;

A_i is an $m_i \times n_i$ matrix .

$M = m_o + m_1 + \ldots + m_L$ is the number of equations of the system which we will assume to be nonredundant and which can be written for short:

$$\textcircled{1}' \quad \begin{cases} \underline{A}x = \underline{b} \\ \gamma x = z(\min) \end{cases} \qquad \text{where } \underline{A} \text{ is an } M \times N \text{ matrix} .$$

$$\boxed{\text{The basic assumption is that } L \gg m_o .}$$

Although we can save an important computational work with respect to the simplex method by applying the decomposition principle [1] to Problem $\textcircled{1}'$ we will, rather, taking advantage of the very special condition $L \gg m_o$, extend the device given by Dantzig and Van Slyke in [2]: From the basis of the system a "working basis" is derived which is considerably smaller in size. Computations are done, as much as possible, on this working basis which result in a substantial savings in computation and storage. A way is given to find the next working basis from iteration to iteration.

In a first part we give definitions and theorems; in a second part we give the general method; and in a third part we give two different ways of updating.

1. SOME DEFINITIONS AND THEOREMS - S_i will refer (depending on context) either to the set of components of x_i or to the set of corresponding columns of the matrix $\underline{A}$.

Suppose we are at some stage of the process and that $\underline{B}$ is a basis for the system. Let

$$\begin{bmatrix} B'_i \\ 0 \\ B_i \\ 0 \end{bmatrix}$$

be the contribution of the set S_i to the basis $\underline{B}$.

THEOREM 1: At least m_i variables from each set S_i $(i = 1,\ldots,L)$ are basic.

Proof: Suppose S_p has less than m_p basic columns. Then there exists a linear combination of rows of B_p which vanishes: (i) $\lambda_p B_p = 0$. Now we can define $\lambda = (0,\ldots,0, \lambda_p, 0,\ldots,0)$ such that $\lambda \underline{B} = \lambda_p B_p = 0$, which contradicts the fact that $\underline{B}$ is nonsingular.

Remark: It follows from (i) that if B_p has exactly m_p basic columns, it is nonsingular.

THEOREM 2: The number of sets S_i containing more than m_i basic variables is at most m_o .

Proof: By the assumption of full rank, each basis has exactly M vectors. By Theorem 1, each set S_i contains at least m_i basic variables; this leaves m_o to make up sets of more than m_i basic variables.

Definition: A set S_i will be said to be essential if it has more than m_i basic variables; the set S_o is always put in the essential class. The other sets will be called inessential.

$$\xi = \{i \mid S_i \text{ is essential}\}$$

$$\bar{\xi} = \{i \mid S_i \text{ is inessential}\} \quad .$$

II. THE GENERAL METHOD - The reduced system is defined by deleting the subsystems $A_i x_i = b_i$ where S_i is inessential. The set of essential basic columns restricted to the reduced system is our working basis B .

The value of inessential basic variables is determined from:

$$B_i x_i^{\prime} = b_i \tag{1}$$

where B_i is square and nonsingular. These smaller systems are readily solved.

THEOREM 3: The working basis is a basis for the reduced system.

Proof: The number of equations in the reduced system equals the number of variables in the working basis since for each inessential set we remove exactly m_i equations and m_i basic columns from the square basis $\underline{B}$. The columns of the working basis are linearly independent since they differ from the columns of the basis $\underline{B}$ by a bunch of zeros in the inessential part, and the columns of $\underline{B}$ are linearly independent.

The size of the working basis B is $m_o + \sum_{i \epsilon \xi} m_i$, considerably smaller than that of $\underline{B}$. Note that the size of B changes from step to step, and that even the number of blocks in B (which is bounded by m_o) changes from step to step.

Suppose we have a basis $\underline{B}$, the columns of which will be denoted $\underline{B}^j$ (j=1,..

Let $B = [B^{h_i}, i = 1,\ldots,|B|]$ be the corresponding working basis.

The problem is:

a) To determine which column enters the basis;

b) To determine which column is dropped from the basis.

1) Let

$\underline{\pi} = (\pi_o, \pi_i, \ldots, \pi_L)$ be the price vector corresponding to $\underline{B}$;

$\pi = (\pi_o, \pi_1 \mid i \epsilon \xi)$ be the price vector corresponding to B ;

$\gamma = (\gamma_o, \gamma_1, \ldots, \gamma_L)$ be the cost vector;

$\underline{c} = (c_o, c_1, \ldots, c_L)$ be the cost vector associated with $\underline{B}$;

$c = (c_i \mid i \epsilon \xi)$ be the cost vector associated with B .

The equation

(2) $$\underline{\pi}\, \underline{B} = \underline{c}$$

defines the price vector. The point is that this equation can be written

(2') $$\pi B = c \quad ;$$

(2'') $$\pi_o B'_i + \pi_i B_i = c_i \qquad i \epsilon \bar{\xi} \quad .$$

We will later give different ways of solving (2'); (2') being solved, substituting π_o in (2'') gives π_i in an easy way.

Let A_σ^s and A'^s_σ denote the sth column of matrices A_σ and A'_σ respectively.

$$\underline{A}_\sigma^s = \begin{pmatrix} A'^s_\sigma \\ 0 \\ A_\sigma^s \\ 0 \end{pmatrix}$$ is the corresponding column of $\underline{A}$.

The column to enter the basis will be given by the usual criterion:

$$\bar{\gamma}^s_\sigma = \gamma^s_\sigma - \underline{\pi}\,\underline{A}^s_\sigma = \min_{\substack{0 \le \tau \le L \\ 1 \le j \le n_\tau}} (\gamma^j_\tau - \underline{\pi}\,\underline{A}^j_\tau) \quad .$$

If $\bar{\gamma}^s_\sigma \geq 0$, we have found an optimal basic solution; if not, we bring $\underline{A}^s_\sigma$ into the basis.

2) $\underline{A}^s_\sigma \in S_\sigma$. If S_σ is essential we have not to worry about the inessential part of the problem which remains unchanged. We simply pivot in the reduced system with the usual criterion for picking the column to drop. We then have to update the right-hand side, and the entering column is the reduced system. That is, to solve

(3) $$B\,\tilde{A}^s_\sigma = A^s_\sigma \quad ;$$

(4) $$B\,\tilde{b} = b \quad .$$

We postpone to a later part the discussion of how (3) and (4) are actually solved. Note that in this case, since B is assumed to be of a reasonable size, the updating is simple.

If S_σ is not essential we have to solve, instead of (3) and (4), the more complicated systems:

(3') $$\underline{B}\,\underline{\tilde{A}}^s_\sigma = \underline{A}^s_\sigma \quad ;$$

(4') $$\underline{B}\,\underline{\tilde{b}} = \underline{b} \quad .$$

We will now prove that solving (3') and (4') is, in fact, equivalent to solving (3) and (4).

Let $\tilde{A}'^s_\sigma$, $\tilde{b}$ denote the part of the updated columns which correspond to the working basis.

$$\begin{bmatrix} B & B'_k & & B'_\sigma & & B'_L \\ & B_k & & & & \\ & & \ddots & & & \\ & & & B_\sigma & & \\ & & & & \ddots & \\ & & & & & B_L \end{bmatrix} \cdot \begin{bmatrix} \tilde A'^s_\sigma \\ \tilde d_k \\ \vdots \\ \tilde A^s_\sigma \\ \vdots \\ \tilde d_L \end{bmatrix}, \begin{bmatrix} \tilde b \\ \tilde b_k \\ \vdots \\ \tilde b_\sigma \\ \vdots \\ \tilde b_L \end{bmatrix} = \begin{bmatrix} A'^s_\sigma \\ 0 \\ 0 \\ A^s_\sigma \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} b \\ b_k \\ \vdots \\ b_\sigma \\ \vdots \\ b_L \end{bmatrix}$$

This can be written:

$$(5)\quad \begin{cases} B_k \quad \tilde d_k = 0 \\ \quad\vdots \qquad\quad = 0 \\ B_\sigma \quad \tilde A^s_\sigma = A^s_\sigma \\ \quad\vdots \qquad\quad = 0 \\ B_L \quad \tilde d_L = 0 \end{cases} \qquad (6)\quad \begin{cases} B_k \quad \tilde b_k = b_k \\ \quad\vdots \\ B_\sigma \quad \tilde b_\sigma = b_\sigma \\ \quad\vdots \\ B_L \quad \tilde b_L = b_L \end{cases} .$$

$$(3'')\quad B\tilde A'^s_\sigma = \begin{bmatrix} A'^s_\sigma \\ 0 \end{bmatrix} - \begin{bmatrix} B'_\sigma \\ 0 \end{bmatrix} \tilde A^s_\sigma \quad ; \quad (4'')\quad B\tilde b = b - \sum_{i\in\bar\xi} \begin{bmatrix} B'_i \\ 0 \end{bmatrix} \tilde b_i \quad .$$

And we see that if we suppose the systems in (5) and (6) are of very low order and that their solution is readily at hand, we have only to solve (3'') and (4''), which are exactly of the same type as (3) and (4).

By the usual simplex criterion we now know which column is to be dropped. Let us suppose that the column to be dropped belongs to the set S_τ . Two cases can occur:

a) S_τ is inessential, so S_τ can only be S_σ . Then the pivoting is only done in the inessential sub-basis B_σ , and the working basis does not change.

b) S_τ is essential, so the method of pivoting is as follows:

<u>Step I</u> - Introduce S_σ in the essential set (the size of the working basis is increased by m_σ) .

<u>Step II</u> - Pivot in the new reduced system.

<u>Step III</u> - If the number of variables in S_τ is now m_τ ,[†] make S_τ inessential (the size of the working basis is decreased by m_τ) .

The algebraic work involved in Equations (2'), (3) and (4) (or (3") and (4")) can be done in essentially two ways. The first is the Revised Simplex Method; the justification of using it is that the working basis will remain of a reasonable size. The second is to solve the system, instead of inverting the matrix B , by a triangularization process which is inspired of [3].

III. <u>UPDATING PROCESS</u> -

A. <u>Revised Simplex Method</u>: Suppose we have an inverse of the working basis B^{-1}; we want to find B^{*-1} where B^* is the basis for the next iteration. If S_σ is essential, we pivot in the reduced system and simply apply the revised simplex algorithm in the reduced system. If S_σ and S_τ are inessential, the working basis does not change. If S_σ is inessential and S_τ is essential, we will describe the three steps outlined above.

[†] If the essential set S_τ had more than $m_{\tau+1}$ basic columns beforehand, this will not be the case.

Step 1. Define

$$\tilde{B} = \left[\begin{array}{c|c} B & \begin{array}{c} B'_\sigma \\ \hline \end{array} \\ \hline 0 & B_\sigma \end{array}\right] \qquad Q = \left[\begin{array}{c} B'_\sigma \\ \hline \end{array}\right] \;;$$

then it is easy to see that:

$$\tilde{B}^{-1} = \left[\begin{array}{c|c} B^{-1} & Q' \\ \hline 0 & B_\sigma^{-1} \end{array}\right] \quad \text{with} \quad Q' = B^{-1} Q B_\sigma^{-1} \; .$$

Step 2. Now we are as in the first case and we use the modified simplex method to get $\tilde{B}^{*-1}$ (where $\tilde{B}^{*}$ is obtained from $\tilde{B}$ by pivoting in the reduced system).

Step 3. If B_τ is now square, we can rearrange the equations and variables in $\tilde{B}^{*}$ so that it looks like this:

$$\tilde{B}^{*} = \left[\begin{array}{c|c} B^{*} & \begin{array}{c} B'_\tau \\ \hline \end{array} \\ \hline & B_\tau \end{array}\right] \qquad \tilde{B}^{*-1} = \left[\begin{array}{c|c} B^{*-1} & Q'_\tau \\ \hline & B_\tau^{-1} \end{array}\right] \; .$$

B^{*} and B^{*-1} are now obtained by dropping the column and rows corresponding to B_τ .

B. Compact Basis Triangularization: Any equations of the type $Bx = b$ or $\pi B = c$ where B is an $m \times m$ nonsingular matrix can be solved by triangularizing B (see [3]). It has been pointed out in [3] that triangularizing may be an efficient method in cases where B has a special structure (such as block angular).

Triangularizing is equivalent to pre-multiplying B by a set of elementary matrices E_i

$$T = E_1, E_2, \dots, E_m B$$

where, in the notation of [3],

T is the triangularized form of B ;

$E_1, \dots, E_m$ is the compact E-structure.

To solve $\pi B = c$ (Equations (2') and (2")), we solve

$$\pi^* T = c$$

and

$$\pi = E_1, E_2, \dots, E_m \pi^* \quad .$$

To solve $Bx = b$ (Equations (3),(4),(5),(6)), we find

$$b^* = E_1, E_2, \dots, E_m b$$

and solve

$$Tx = b^* \quad .$$

Having determined S_σ and S_τ , we now show the steps in pivoting.

If S_σ is essential, we pivot in the reduced working basis. If S_σ and S_τ are inessential, we pivot in the smaller basis B_σ . If S_σ is inessential and S_τ essential, then we show the three steps.

Let T_σ be the triangularized form of B_σ , E_σ its compact E-structure, and J the permutation matrix used to triangularize B . Let T be the triangularized form of B , and E its compact E-structure.

Step 1. Introducing S_σ into the essential class, we modify the working basis by introducing B_σ into it. We get[†]

$$\text{New } T = \begin{array}{|c|c|}\hline T_\sigma & 0 \\ \hline EJQ & T \\ \hline \end{array} \qquad Q = \begin{array}{|c|}\hline B'_\sigma \\ \hline 0 \\ \hline \end{array}$$

$$\text{New } E = \begin{array}{|c|c|}\hline E_\sigma & 0 \\ \hline 0 & E \\ \hline \end{array}$$

Step 2. Apply the pivot operation in the new B .

Step 3. Assume ξ_τ has become inessential; to remove this block from the working basis, assume the columns of S_τ in B lie between i^{th} and $i+m_\tau$ th column. Define

$$T = \begin{array}{ccc|c} & i & i+m_\tau & \\ \hline T_1 & & 0 & i \\ & T_2 & & i+m_\tau \\ T_4 & & T_3 & \\ \hline \end{array} \qquad E = \begin{array}{ccc|c} & i & i+m_\tau & \\ \hline E_1 & & E_4 & i \\ & E_2 & & i+m_\tau \\ & & E_3 & \\ \hline \end{array}$$

Then

[†] Note: If we introduce the inessential set S_σ at the top LHS of the T and E , we always obtain the new E structure in this manner.

triangularized form of B , and E its compact E-structure.

Step 1. Introducing S_σ into the essential class, we modify the working basis by introducing B_σ into it. We get[†]

$$\text{New } T = \left[\begin{array}{c|c} T_\sigma & 0 \\ \hline EJQ & T \end{array}\right] \qquad Q = \left[\begin{array}{c} B'_\sigma \\ \hline 0 \end{array}\right]$$

$$\text{New } E = \left[\begin{array}{c|c} E_\sigma & 0 \\ \hline 0 & E \end{array}\right]$$

Step 2. Apply the pivot operation in the new B .

Step 3. Assume ξ_τ has become inessential; to remove this block from the working basis, assume the columns of S_τ in B lie between i^{th} and $i+m_\tau{}^{th}$ column. Define

$$T = \begin{array}{c|c|c|l} \multicolumn{1}{c}{} & \multicolumn{1}{c}{i} & \multicolumn{1}{c}{i+m_\tau} & \\ \hline T_1 & & 0 & i \\ \hline & T_2 & & i+m_\tau \\ \hline T_4 & & T_3 & \\ \hline \end{array} \qquad E = \begin{array}{c|c|c|l} \multicolumn{1}{c}{} & \multicolumn{1}{c}{i} & \multicolumn{1}{c}{i+m_\tau} & \\ \hline E_1 & & E_4 & i \\ \hline & E_2 & & i+m_\tau \\ \hline & & E_3 & \\ \hline \end{array} .$$

Then

[†] Note: If we introduce the inessential set S_σ at the top LHS of the T and E , we always obtain the new E structure in this manner.

$$\text{new } T = \begin{array}{|c|c|}\hline T_1 & 0 \\ \hline T_4 & T_3 \\ \hline \end{array} \qquad \text{new } E = \begin{array}{|c|c|}\hline E_1 & E_4 \\ \hline & E_3 \\ \hline \end{array}$$

and $$T_T = T_2 \qquad\qquad E_T = E_2 \quad .$$

In conclusion, it seems that Method A may be preferred if the working basis is of small size and if the process of the revised simplex algorithm on it is not too long, as Method B will be preferred if the working basis itself is of large dimension.

REFERENCES

[1] Dantzig, G. B., Linear Programming and Extensions, Princeton University Press, Princeton, New Jersey, Chapter 23 (1963).

[2] Dantzig, G. B., and R. M. Van Slyke, Generalized Upper Bounding Techniques for Linear Programming - I," Operations Research Center 64-17, University of California, Berkeley (March 1965).

[3] Dantzig, G. B., "Compact Basis Triangularization for the Simplex Method," Recent Advances in Mathematical Programming, P. Wolfe and R. L. Graves, Eds., pp. 125-132, McGraw-Hill (1963).

[4] Bennett, John, "An Approach to Some Structured Linear Programming Problems," Basser Computing Department, School of Physics, University of Sydney, Australia (March 1963).

Unclassified
Security Classification

DOCUMENT CONTROL DATA - R&D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)*
University of California (Berkeley)

2a. REPORT SECURITY CLASSIFICATION
Unclassified

2b. GROUP
NA

3. REPORT TITLE
An Extension of Generalized Upper Bounded Techniques I for Structured Linear Programs

4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*
Technical Report - December 1965

5. AUTHOR(S) *(Last name, first name, initial)*
Sakarovitch, M.
Saigal, R.

6. REPORT DATE December 1965	7a. TOTAL NO. OF PAGES 12	7b. NO. OF REFS 4

8a. CONTRACT OR GRANT NO.
DA-31-124-ARO-D-331

b. PROJECT NO.
20014501B14C

c.

d.

9a. ORIGINATOR'S REPORT NUMBER(S)
ORC 65-33

9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)*
5307.7

10. AVAILABILITY/LIMITATION NOTICES
Qualified requesters may obtain copies of this report from DDC.

11. SUPPLEMENTARY NOTES
None

12. SPONSORING MILITARY ACTIVITY
U. S. Army Research Office-Durham
Box CM, Duke Station
Durham, N. C. 27706

13. ABSTRACT

An algorithm is developed for solving a special structured linear program. The particular structure studied has a large number of blocks coupled together by a relatively few connecting equations. The method proposed is an extension of "Generalized Upper Bounding Techniques for Linear Programming-I," by G. B. Dantzig and R. M. Van Slyke (March 1965) and, from the basis, defines a working basis which is much smaller in size than the original. Two methods of updating the working basis are proposed.

Unclassified
Security Classification

Unclassified
Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Structured Linear Programs Algorithms						

INSTRUCTIONS

1. ORIGINATING ACTIVITY: Enter the name and address of the contractor, subcontractor, grantee, Department of Defense activity or other organization *(corporate author)* issuing the report.

2*a*. REPORT SECURITY CLASSIFICATION: Enter the overall security classification of the report. Indicate whether "Restricted Data" is included. Marking is to be in accordance with appropriate security regulations.

2*b*. GROUP: Automatic downgrading is specified in DoD Directive 5200.10 and Armed Forces Industrial Manual. Enter the group number. Also, when applicable, show that optional markings have been used for Group 3 and Group 4 as authorized.

3. REPORT TITLE: Enter the complete report title in all capital letters. Titles in all cases should be unclassified. If a meaningful title cannot be selected without classification, show title classification in all capitals in parenthesis immediately following the title.

4. DESCRIPTIVE NOTES: If appropriate, enter the type of report, e.g., interim, progress, summary, annual, or final. Give the inclusive dates when a specific reporting period is covered.

5. AUTHOR(S): Enter the name(s) of author(s) as shown on or in the report. Enter last name, first name, middle initial. If military, show rank and branch of service. The name of the principal author is an absolute minimum requirement.

6. REPORT DATE: Enter the date of the report as day, month, year; or month, year. If more than one date appears on the report, use date of publication.

7*a*. TOTAL NUMBER OF PAGES: The total page count should follow normal pagination procedures, i.e., enter the number of pages containing information.

7*b*. NUMBER OF REFERENCES: Enter the total number of references cited in the report.

8*a*. CONTRACT OR GRANT NUMBER: If appropriate, enter the applicable number of the contract or grant under which the report was written.

8*b*, 8*c*, & 8*d*. PROJECT NUMBER: Enter the appropriate military department identification, such as project number, subproject number, system numbers, task number, etc.

9*a*. ORIGINATOR'S REPORT NUMBER(S): Enter the official report number by which the document will be identified and controlled by the originating activity. This number must be unique to this report.

9*b*. OTHER REPORT NUMBER(S): If the report has been assigned any other report numbers *(either by the originator or by the sponsor)*, also enter this number(s).

10. AVAILABILITY/LIMITATION NOTICES: Enter any limitations on further dissemination of the report, other than those imposed by security classification, using standard statements such as:

(1) "Qualified requesters may obtain copies of this report from DDC."

(2) "Foreign announcement and dissemination of this report by DDC is not authorized."

(3) "U. S. Government agencies may obtain copies of this report directly from DDC. Other qualified DDC users shall request through
_______________________________."

(4) "U. S. military agencies may obtain copies of this report directly from DDC. Other qualified users shall request through
_______________________________."

(5) "All distribution of this report is controlled. Qualified DDC users shall request through
_______________________________."

If the report has been furnished to the Office of Technical Services, Department of Commerce, for sale to the public, indicate this fact and enter the price, if known.

11. SUPPLEMENTARY NOTES: Use for additional explanatory notes.

12. SPONSORING MILITARY ACTIVITY: Enter the name of the departmental project office or laboratory sponsoring *(paying for)* the research and development. Include address.

13. ABSTRACT: Enter an abstract giving a brief and factual summary of the document indicative of the report, even though it may also appear elsewhere in the body of the technical report. If additional space is required, a continuation sheet shall be attached.

It is highly desirable that the abstract of classified reports be unclassified. Each paragraph of the abstract shall end with an indication of the military security classification of the information in the paragraph, represented as *(TS)*, *(S)*, *(C)*, or *(U)*.

There is no limitation on the length of the abstract. However, the suggested length is from 150 to 225 words.

14. KEY WORDS: Key words are technically meaningful terms or short phrases that characterize a report and may be used as index entries for cataloging the report. Key words must be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location, may be used as key words but will be followed by an indication of technical context. The assignment of links, rules, and weights is optional.

Unclassified
Security Classification